

Chapter 4

The EON model

4.1 Overview

The present thesis is focused in the development of a generic scheduling framework applicable to the chemical industry. This chapter describes the main innovation aspects developed in this work: the timing model. In this chapter, the Event Operation Network (EON) model will be described in detail and two important topics applicable to this kind of industry will be addressed. The first one is the resources availability and the second one the intermediate storage.

As in other kinds of industry, chemical plants are characterized by the consumption of resources other than units. Steam, electricity and manpower are typical resources used in this kind of processes, other resources like raw materials should be also taken into account. This chapter will show how these resources are calculated and taken into account using the EON model.

The intermediate storage problem, as a key issue to be solved within the chemical industry, should also be addressed. The traditional definitions of the constraints associated to the intermediate storage (ZW, IS, UIS, etc) are not really well suited when used in a detailed model as EON. In this chapter a new modeling of the IS policies addressed to the real situations encountered in the chemical industry will be explained and applied to the EON model.

4.2 Description of the Event operation Network (EON)

Despite the general recipe description which is very useful to fully describe all the recipes using an appropriate user interface, the rigorous modeling of the operation timing involved in the recipe is achieved by means of a few elements integrating the Event Operation Network (EON). These elements are events, operations and links. Events designate these time instants in which some change occur. They are represented by nodes in the EON graph and should be linked to operations and can be linked also with other events (fig 4.1). Each event n is associated to a time value, T_n , and a lower bound T_n^{min} (Eqn 4.4 on page 41). The lower

Chapter 4. The EON model

bound can be used to force delays on a given solution in order to force a new solution that reaches special constraints (f.e. calendar constraints) The times values T_n for each event will be the variables of the timing problem we want to solve.

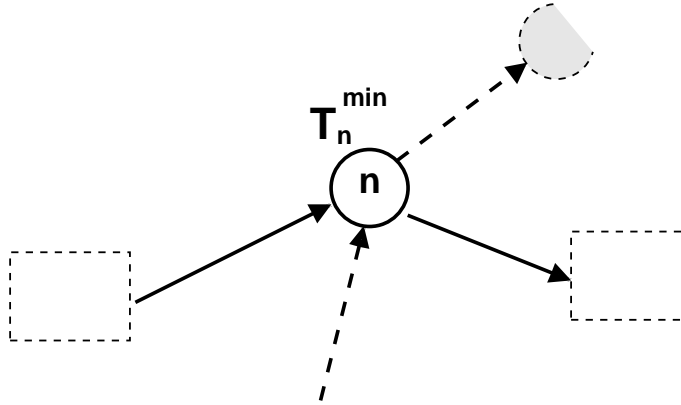


Figure 4.1: Events in the EON model

Operations designate those time intervals to be observed between the starting event (initial node, IN) and the termination event (final node, FN). Each operation m is represented by a box linked with solid arrows to its associated nodes (IN_m and FN_m). Operations are the equality links between nodes given by equation 4.5 in terms of the characteristic properties of each operation: the operation time TOP and the waiting time. The operation time is a pre-determined value (as a function of amount being processed, unit assigned, product changeover, or a value referenced to another operation or to auxiliary equipment unit) which origin does not affect the EON description. The waiting time is the time slack for each operation bounded as expressed by equation 4.6.

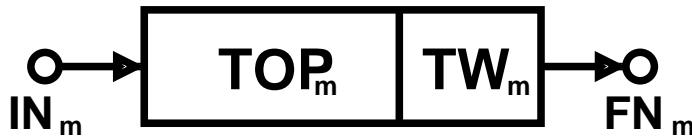


Figure 4.2: Operation of the EON Model

Finally, links designate those explicit internodal (event-event) precedence constraints. Each link k is represented by a dashed arrow from its origin node (ON_k) to its destiny node (DN_k) and its associated increment of time (ΔT_k). The inequality constraint given by internodal links is expressed by equation 4.7:

Up to this point, the EON representation allows to describe all the timing constraints to determine a feasible timing for the structure of activities of a given process. However, an objective is needed to discriminate a good solution among the infinite feasible ones. A simple

4.3. Building Event Operation Networks

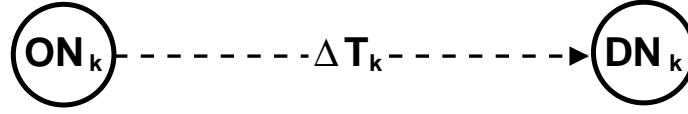


Figure 4.3: Links between events in the EON model

and practical one is makespan which allows to compress the resulting schedules. However, makespan (equation 4.1) may still be a very degenerated objective. A more flexible and general objective may be defined by the weighted contributions of makespan, waiting times and starting times (equation 4.2).

$$Z = MS \quad (4.1)$$

So the whole LP:Problem formulation may be defined with the following set of equations: minimize:

$$Z = C^1 \cdot MS + \sum_{m=1}^M C_m^2 \cdot TW_m + \sum_{n=1}^N C_n^3 \cdot T_n \quad (4.2)$$

subject to:

$$0 \leq MS \geq T_n \quad \forall n; n = 1 \dots N \quad (4.3)$$

$$0 \leq T_n \geq T_n^{min} \quad \forall n; n = 1 \dots N \quad (4.4)$$

$$T_{FN_m} - T_{IN_m} - TOP_m = TW_m \quad \forall m; m = 1 \dots M \quad (4.5)$$

$$0 \leq TW_m \leq TW_m^{max} \quad \forall m; m = 1 \dots M \quad (4.6)$$

$$T_{DN_k} \geq T_{ON_k} + \Delta T_k \quad \forall k; k = 1 \dots K \quad (4.7)$$

The main advantage of the EON model is that is composed with very simple pieces which results in a very simple model, however it allows to describe very complex production recipes with complex time constraints. Solving the related LP a good starting point is found (solver or strategy).

4.3 Building Event Operation Networks

4.3.1 Building basics

The EON structure allows to build complex recipes automatically from the simpler description as process-stage-operation. Each stage is composed by a certain number of operations to

Chapter 4. The EON model

perform consecutively this situation is translated directly to the EON description as a group of operations which share initial and ending events as it is shown in figure 4.4.

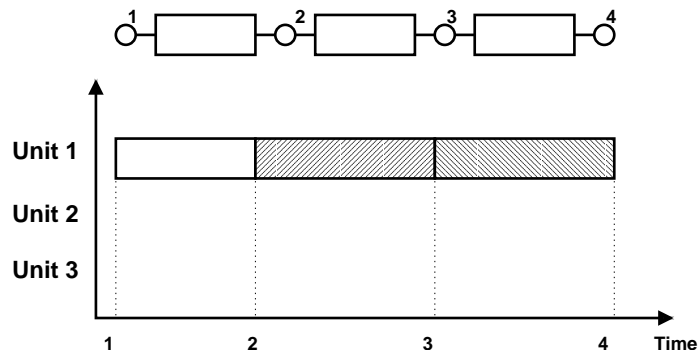


Figure 4.4: EON representation of one stage with three operations

However, the same EON representation can also be used to represent three different stages with a Zero wait (ZW) Policy.(Fig. 4.5) This point illustrates one of the main characteristics of the EON networks: the EON representation is a graphical representation of the time constraints associated to the problem, so only time constraints are represented.

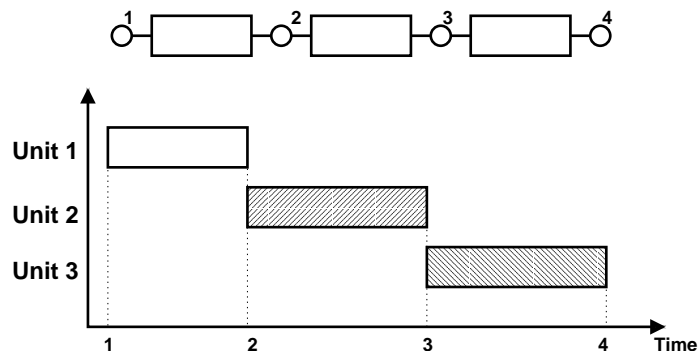


Figure 4.5: Three stages of one operation with ZW policy

The basic methodology to build EON networks for one recipe can be summarized in two main steps:

1. Create one operation for each one described in the process-stage-operation description
2. Convert the links declared in the process-stage-operation description to a EON representation.

The first point can be automatically performed as well as each operation in the process-stage-operation representation can be represented as one operation in the EON graph.

4.3. Building Event Operation Networks

The second point requires a more detailed explanation in the following section.

4.3.2 Standard Links

Some pre-defined links has been established to allow an easier description of the recipe by the user. Each of them has an specific way of being transfered to an EON structure. These links can be categorized in two groups:

- Equality links
 - Simultaneous
 - Consecutive
 - Relationship between initial times
 - Relationship between ending times
- Inequality links
 - Initial times precedence
 - Ending times precedence
 - Consecutive precedence
 - Time inclusion

These links stablish different constraints between the initial time (IT) and the ending time (ET) of the first operation and the initial and ending times (IT' and ET') of the second operation as it is described below in detail.

4.3.2.1 Simultaneous link

This kind of link is expressed by equations 4.8 and 4.9. As its name indicates, this link forces the simultaneity of the two operations. The translation of this constraints into EON is obvious as it means that the initial end ending evens of two operations are the same event. This translation is shown in figure 4.6.

$$IT = IT' \quad (4.8)$$

$$ET = ET' \quad (4.9)$$

4.3.2.2 Consecutive Link

A consecutive link is characterized by the equation 4.10. It represents the classical relationship between stages in a workshop. Its translation into an EON representation requires the introduction of an additional waiting operation in the EON. The operation time is the K value of the constraint associated to this kind of link. The EON resulting of this kind of link is represented in figure 4.7

$$IT' = ET + K \quad (4.10)$$

Chapter 4. The EON model

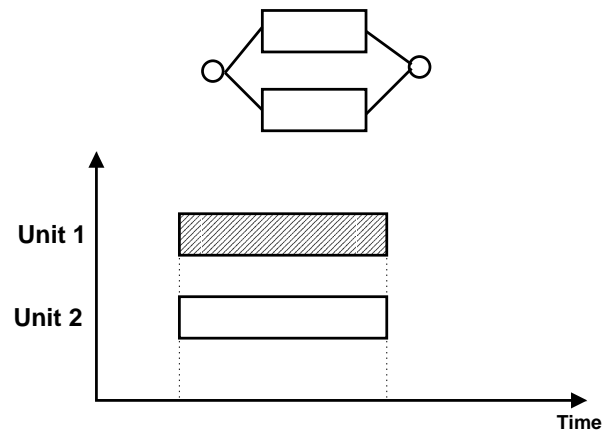


Figure 4.6: Simultaneous link into EON graphical representation.

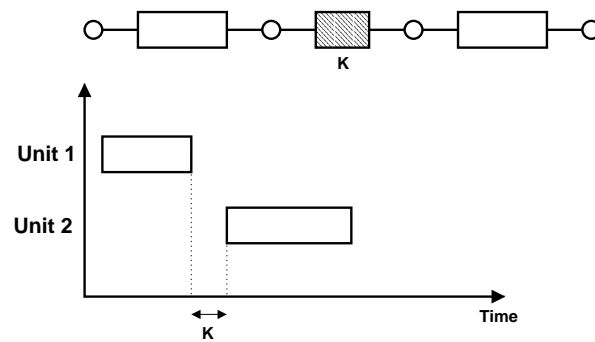


Figure 4.7: EON representation of a consecutive link

4.3. Building Event Operation Networks

4.3.2.3 Initial times Link

This kind of link is characterized with the constraint 4.11. It represents a situation where an operation should start after another one is started. The gantt chart and the EON associated to this constraint is represented in figure 4.8.

$$IT' = IT + K \quad (4.11)$$

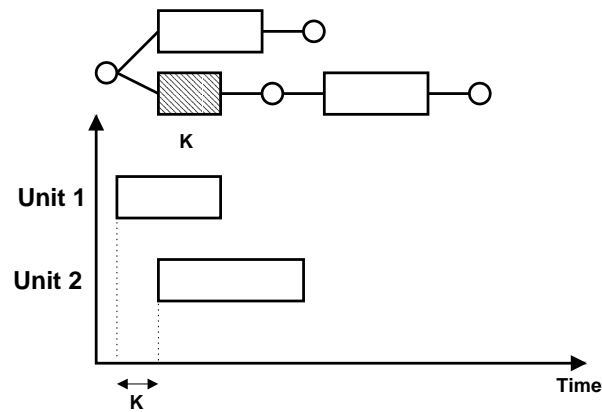


Figure 4.8: EON representation of a relationship between initial times link

4.3.2.4 Ending times link

This kind of link is characterized with the constraint 4.12. It represents a situation where an operation cannot be started before someone else. The gantt chart and the EON associated to this constraint is represented in figure 4.9.

$$ET' = ET + K \quad (4.12)$$

4.3.2.5 Initial times precedence

This kind of link is characterized with the constraint 4.13. It represents a situation where an operation should start sometime after another one is started. The gantt chart and the EON associated to this constraint is represented in figure 4.10.

$$IT' \geq IT + K \quad (4.13)$$

Chapter 4. The EON model

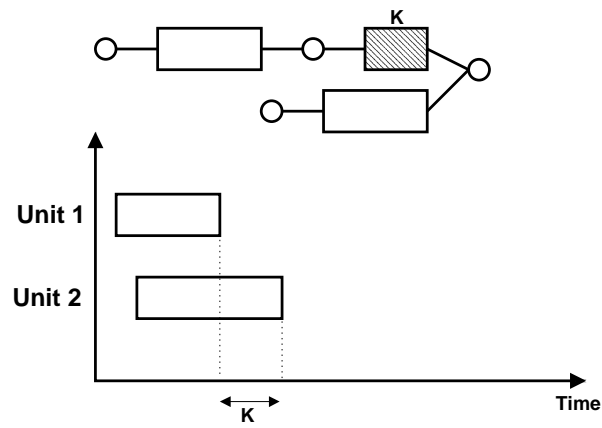


Figure 4.9: EON representation of a relationship between ending times link.

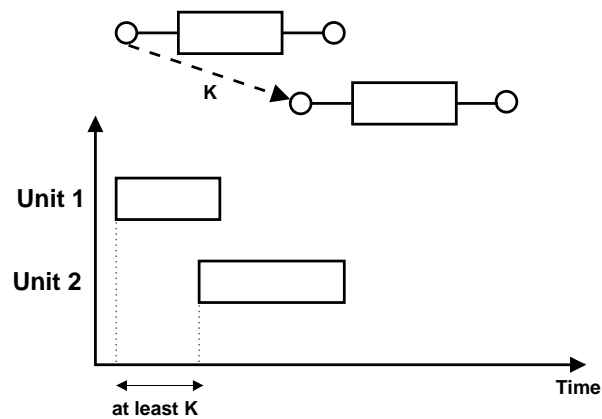


Figure 4.10: EON representation of initial times precedence link.

4.3. Building Event Operation Networks

4.3.2.6 End times precedence

This kind of link is characterized with the constraint 4.14. It represents a situation where an operation must end sometime after another one is ended. The gantt chart and the EON associated to this constraint is represented in figure 4.11.

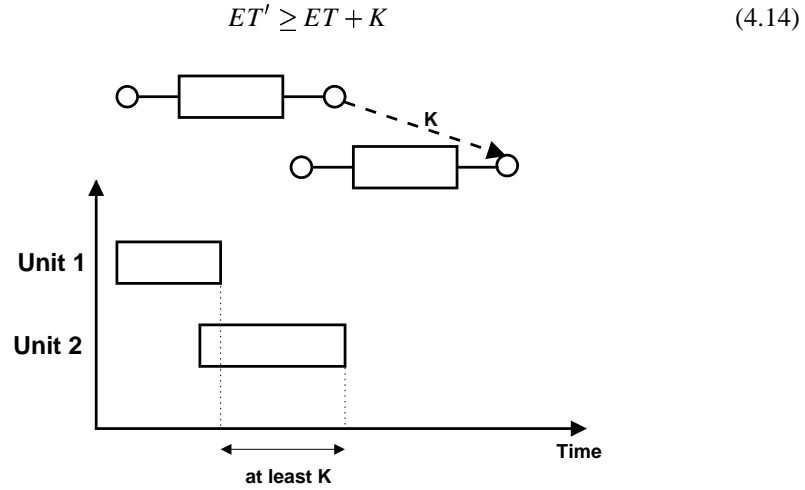


Figure 4.11: EON representation of ending times precedence link.

4.3.2.7 Consecutive precedence

This kind of link expresses the inequality counterpart of the consecutive link as is shown by the associated constraint (4.15). As an inequality link, can be also described as a link between events as is shown in figure 4.12.

$$IT' \geq ET + K \quad (4.15)$$

4.3.2.8 Time inclusion Link

This is an special kind of link. The meaning of this link is that an operation should be performed within another operation. The constraints associated are 4.16 and 4.17. The aspect of the resulting EON graft is presented in figure 4.13.

$$IT' \geq IT \quad (4.16)$$

$$ET \geq ET' \quad (4.17)$$

All the links presented allow to the potential user to build recipes of complex structure using the simple elements of the EON model. In fact, the user does not need to know the

Chapter 4. The EON model

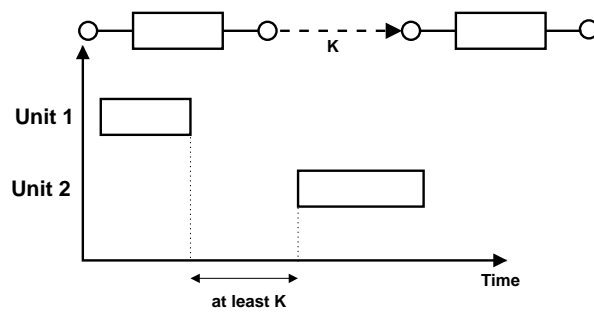


Figure 4.12: EON representation of consecutive precedence

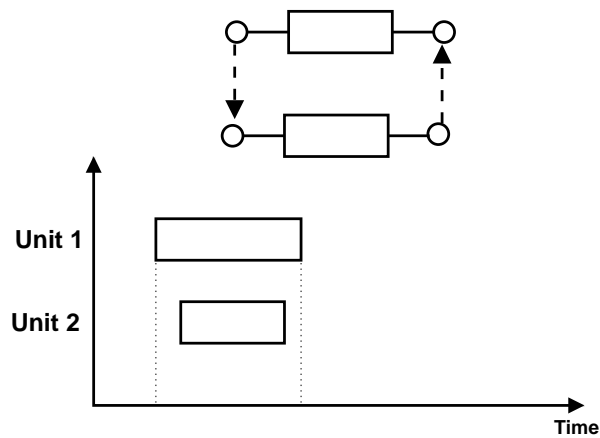


Figure 4.13: EON representation of a time inclusion link

4.3. Building Event Operation Networks

internal EON representation as the EON graph can be build easily from the recipe data model presented previously.

4.3.3 Multiple batches

The same modeling elements and tools used up to this point for describing recipes may be used to describe production schedules as well. Obviously, a production schedule may be conceived as a large recipe whose complexity may not necessarily be higher than that of its constituting pieces. Therefore, node to node links may be used to force those operations executed in the same equipment unit to do not overlap. Figure 4.14 shows the EON representation corresponding to a production schedule in which two jobs following the same process recipe have been sequentially dispatched. The links from nodes 2,3 and 4 to nodes 5,6 and 7 force those times associated to the second batch to be after than the ending time of the former ones.

This kind of links can be easily generated automatically once the sequence of operations to be performed in each equipment has been defined. In the case of the approach adopted in this work such information is derived from a sequence of jobs following their assigned process recipes and paths. Hence, for every schedule proposed (either manual or automatic) the corresponding EON representation is generated merging the EON representations of the recipes involved.

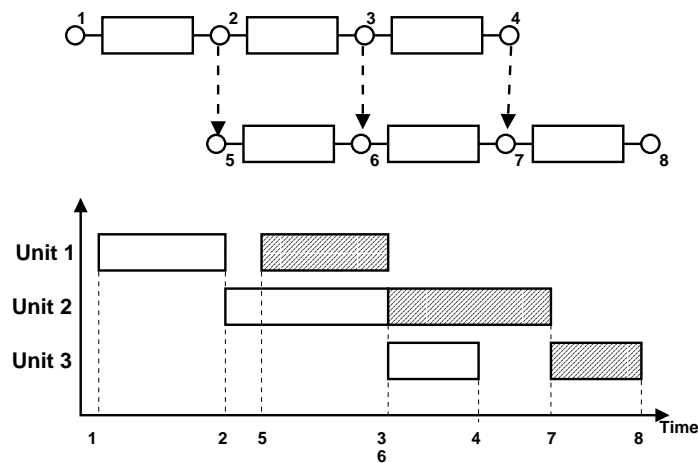


Figure 4.14: links between two batches

Sometimes, this general non-overlapping constraint can be skipped. In this cases the unit default time constraint is deactivated allowing the overlapping of different operations in the same unit. This kind of units allows the analysis of the effects of adding new units to an existing plant without defining new recipes. An example of the use of this kind of units is shown in figure 4.15.

On other cases, a minimum time gap between two tasks that are performed in the same

Chapter 4. The EON model

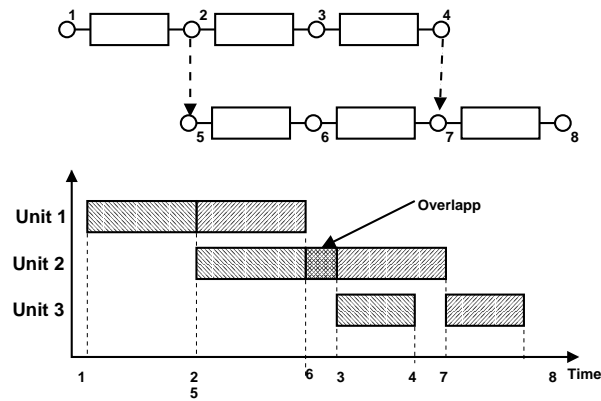


Figure 4.15: Effect on the gantt and the EON of deactivating default time constraint in Unit 2.

unit can be easily forced including a value of ΔT_k in the corresponding link as is it shown in figure 4.16.

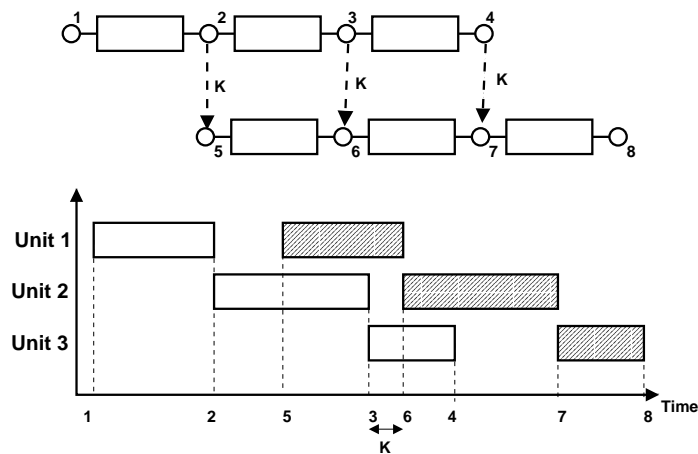


Figure 4.16: Effect of introducing gaps between different tasks

As it is shown, the EON modeling is useful not only to describe complex time relationships between operations in the same recipe, but also complex time relationships between different batches. This flexibility allows to determine the timing of a whole schedule using the simple formulation presented in page 41 once the whole problem is defined as an EON structure. However, the generation of the EON structure, that can be very complex for the whole problem, can be easily automated using the rules presented in this section from the data model introduced in the previous chapter.

4.4. Shared resources

Additionally, more constraints can be included in the EON structure to include advanced constraints as storage constraints that will be shown in detail in a separate chapter.

4.4 Shared resources

One of the characteristics of the EON model described in the previous chapter is that any node can have an associated minimum time. This characteristic is used to take into account the constraints associated to the shared resources. The strategy based in this work to take into account that constraint is based in the work of Graells (1995) adapted to its use within a EON framework. The goal of the algorithm is to incorporate the resource constraints into the calculation of the operation timing adjusting the consumption of a given resource to the resource availability for that resource. The resource availability is given as an availability vs. time profile as shown in figure 4.17.

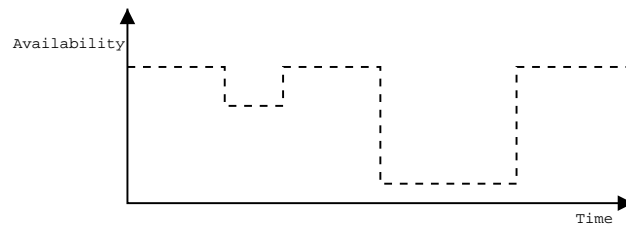


Figure 4.17: Availability for one resource

The algorithm used is shown in figure 4.18. Basically consists on an iterative procedure that contains three steps: timing, calculation of resource profiles and adjustment of the resource profiles. The initial point is the calculation of the timing. EON model as explained in the previous chapter is used for this task.

The calculation of the resource profiles for a given timing is performed as the added contribution to the resource consumption for each batch. Each batch generates its own resource consumption profiles (figure 4.19) then, all the individual contributions are added to generate the global resource consumption profile for the whole schedule (see figure 4.20).

The following step is to detect the first violation of the availability profile for the resource. At this point the contributing operations are detected and one of them is chosen according the following criteria:

1. The contributing operations are ordered in increasing order according the sequence of batches.
2. The different contributions are added one by one to detect the first contribution that exceeds the availability of the resource.
3. The operation associated to this contribution is the one selected to be shifted (see figure 4.21)

Chapter 4. The EON model

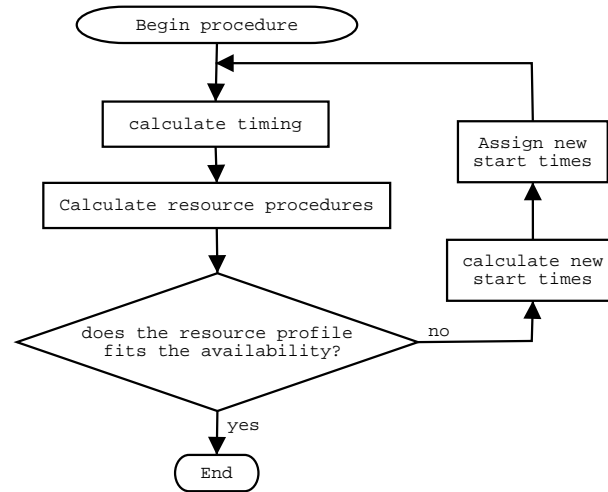


Figure 4.18: De-overlapping algorithm

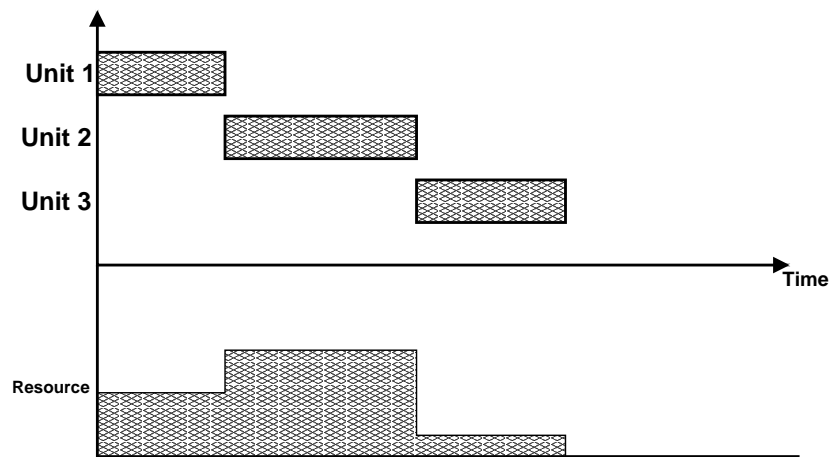


Figure 4.19: resource consumption for one batch

4.4. Shared resources

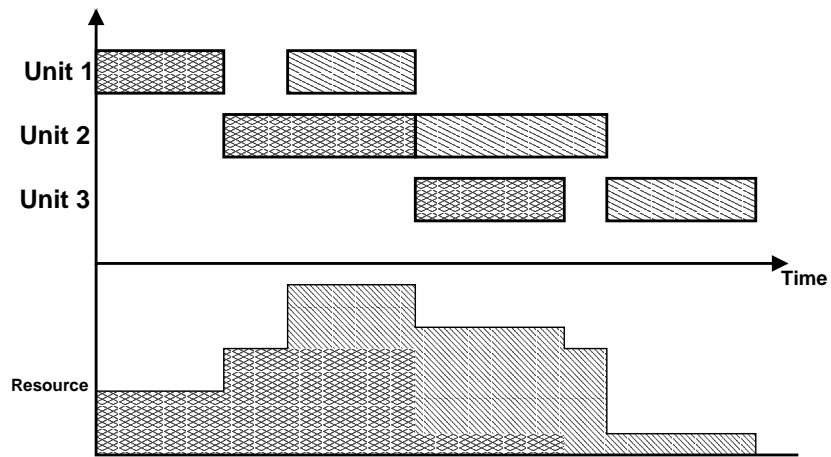


Figure 4.20: resource consumption for two batches

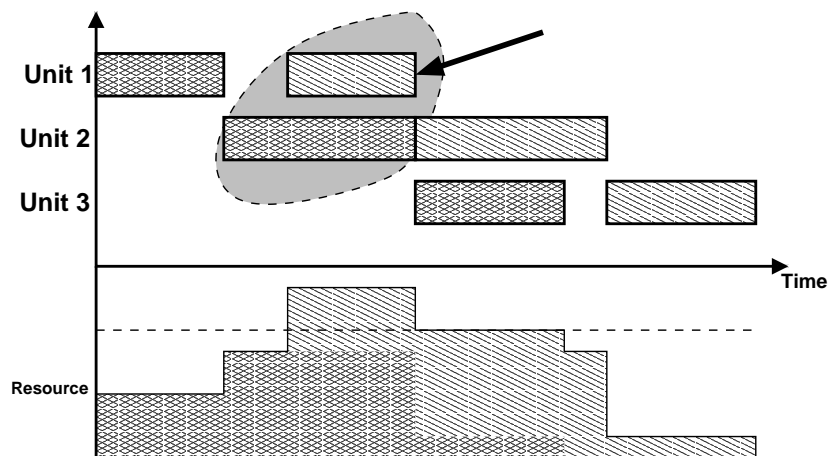


Figure 4.21: Detection of the contributing operations

Chapter 4. The EON model

Once the operation to be shifted is selected, the shifting time is calculated. Typically, the necessary shifting time is calculated taking into account:

- The cumulative profile for the contributing operations that are previous in the list of batches.
- The availability profile of the resources implied in the calculation.
- The amount of the contribution to the resource consumption for the operation to be shifted.

The ending times of the contributions corresponding to the complementary contributing operations gives the initial candidates for the shifting time. These times are associated to the different steps given in the profile. The shifting time correspond which the minimum of those times(steps) that allows an increase of the consumption at least equal to the contribution of the operation to be shifted without violating the availability profile.

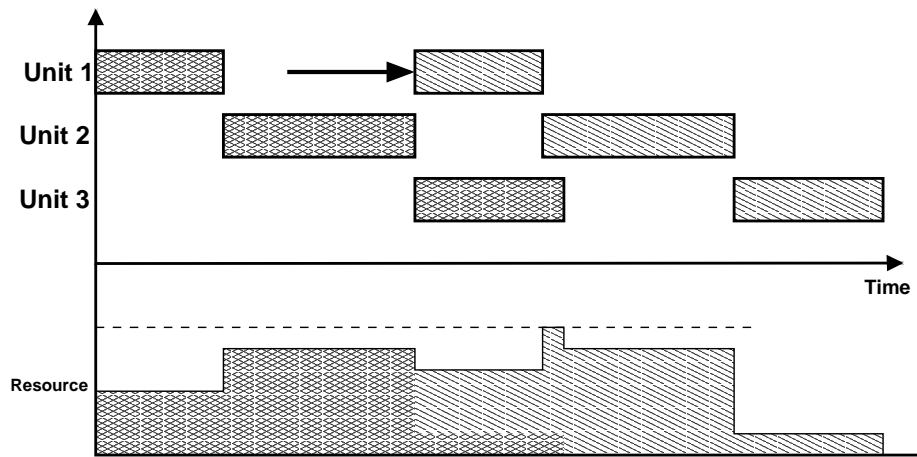


Figure 4.22: Deoverlapping performed

Once determined the shifting time, the start event of the EON associated to the operation to be shifted is modified assigning a new minimum time and the EON is recalculated to find out the new operation times as shown in figure 4.22.

In some situations the remaining availability profile shows that not enough amount of resource is available to fit the resource consumption of the operation to be shifted. In those situations, the approach used is to “soften” the constraint resource, deactivating the deoverlapping for that resource and reporting it to the solution. In such cases the timing problem is infeasible, but a valid schedule out of infeasibilities is presented showing the cause of the infeasibility to the user. This approach is useful in the production scenarios as the user can adopt the necessary measures to modify the availability of the resource (i.e. asking the workers for overtime hours).

4.5. Storage management

4.5 Storage management

Traditionally, several storage policies can be conceived: no intermediate storage (NIS), which can be divided in Finite wait (FW) and zero wait (ZW), and also intermediate storage (IS), which can be either unlimited (UIS) or finite (FIS).

The case of finite storage, defined in the same recipe as a wait time, can be defined for each operation.

The case of intermediate storage is very relevant in the chemical industry where there are usually limitations in the tanks. The UIS case can be considered as an FIS with a very high capacity. So the real problem is the modeling of the finite intermediate storage.

Nevertheless, the description of an storage as FIS is very simplistic. In real plants is often found that the situation is very different from this description. In fact, according to the specific plant, a more detailed description of the reality is needed. The present thesis deals with a more realistic classification of the intermediate storage. A key issue is that intermediate storage has the associated constraints of minimum and maximum level that can not be violated. According to the use of the intermediate storage, it can be classified as:

- Dedicated: the storage is only used by one material.
- Shared: the storage can be used by more than one material at the same time (typically a warehouse)
- Shared Exclusive: the storage can be used along the time by different materials but only one material can be in the storage at the same time.

According to the operation limitations associated to the storage it is classified as:

- Single inlet outlet (SIO): In this type of storage only an inlet or an outlet is considered at the same time. This kind of storage is useful when the time of charge and discharge is negligible in front of the processing time or even is considered zero.
- Single inlet single outlet (SISO): In this type of storage a charge and a discharge can be simultaneous. However, two charges or two discharges cannot be simultaneous. An example of this situation is a storage that only has one pump in the intake and other in the outlet.
- Multiple inlets and multiple outlets (MIMO): In this type of storage, several charges and discharges may be simultaneous. This is the most general description of storage.
- Single inlet and multiple outlets (SIMO): This type is a combination of the SISO and the MIMO two charges cannot be simultaneous, but a charge can be simultaneous with several discharges.
- Multiple inlet and single outlet (MISO): it is the complementary to the previous one, two discharges cannot be simultaneous, but a discharge can be simultaneous with several charges.

Each storage is characterized with:

Chapter 4. The EON model

- An initial level for each one of the materials associated to this storage.
- A sequence of level variations for each one of the materials associated to this storage.

Once the constraints associated to a schedule have been described as an EON, each storage has a sequence of volume variations. According to the use of the storage the following aspects should be taken into account to solve the problem:

- If we have a dedicated storage for one material, all the volume variations of this material belong to this storage.
- If we have a shared storage for a list of materials, all the variations of volume of these materials belong to this storage.
- If we have a shared exclusive storage for a list of materials, all the variations of volume of these materials belong to this storage. For such a case, additional links must be introduced for each product changeover to guarantee that the next variation after the changeover will start later in time than all those previous in the variation sequence.

The operation limitations should lead to consider how can the capacity constraints be transformed into time constraints that can be added to the EON network.

4.5.1 SIO: single input output.

For a given storage and given a sequence of level variations. V_i $1 \leq i \leq N$. The schedule will be into the storage limits if the following expressions are true:

$$I + \sum_{i=0}^n V_i \leq MAX \quad \forall n, 0 \leq n \leq N \quad (4.18)$$

$$I + \sum_{i=0}^n V_i \geq MIN \quad \forall n, 0 \leq n \leq N \quad (4.19)$$

Additionally, time constraints should be added to the EON to force the variations of volume to be in the given sequence.

$$T_{NI_{V_n}} \geq T_{NF_{V_{n-1}}} \quad \forall n, 2 \leq n \leq N \quad (4.20)$$

These constraints determine an EON model represented in figure 4.23.

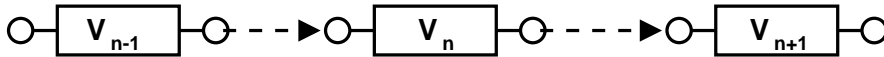


Figure 4.23: EON with SIO constraints

4.5. Storage management

4.5.2 SISO: Single Input Single Output

In this case the first step is to divide the sequence of variations as a sequence of charges and discharges. Considered alone, the aspect of the variation for each of the two contributions is shown in figure 4.24.

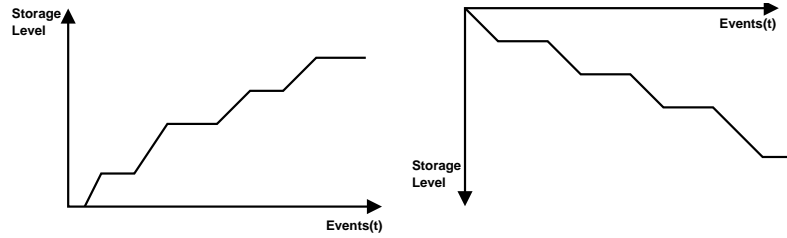


Figure 4.24: Charge and discharge profiles

An important aspect to be considered is that while the general profile of the contributions to the charge and to the discharge of the storage is known, the exact time when a charge or a discharge starts or finishes is an unknown variable. Nevertheless, for each level variation the start and ending events of the corresponding EON are known, as is also known the amount of the variation and the time that this variation will last. The problem is reduced to find the points of each profile where the storage limits are from the other profile defining the storage working range. This is shown in figure 4.25.

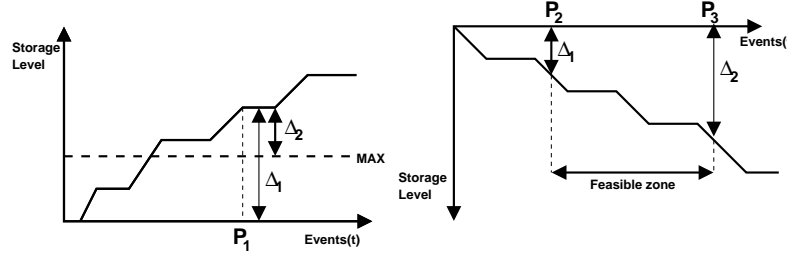


Figure 4.25: Finding a feasible zone for an end of a charge

According to figure 4.25, the event corresponding to the point P_1 should be between P_2 and P_3 to insure that the addition of the contributions of the charge and the discharge will be between the limits of the storage. The problem can be generalized as follow:

Given:

A sequence of N charges: C_i ; $1 \leq i \leq N$

And a sequence of M discharges: D_j ; $1 \leq j \leq M$

The contribution of the sequence of charges at every point can be expressed as the addition of previous $n - 1$ contributions plus a partial contribution of the charge n as is expressed in the following equation:

Chapter 4. The EON model

$$\sum_{i=0}^{n(t)-1} C_i + K(t) \cdot C_{n(t)} \quad 0 \leq K(t) \leq 1 \quad (4.21)$$

In the same way, the contribution of a sequence of discharges at any time can be expressed as:

$$\sum_{j=0}^{m(t)-1} D_j + K'(t) \cdot D_{m(t)} \quad 0 \leq K'(t) \leq 1 \quad (4.22)$$

So in order to keep the profile level between maximum and minimum levels links must be established between the charge sequence and the discharge sequence so that each charge n cannot end before a contribution of discharges has been made according to the expression:

$$I + \sum_{i=1}^n C_i - \sum_{j=0}^{\mu_n-1} D_j - K_n \cdot D_{\mu_n} = MAX \quad 0 \leq K_n \leq 1 : 0 \leq \mu_n \leq M \quad (4.23)$$

and also cannot end after the contribution of discharges given by the following expression has been done:

$$I + \sum_{i=1}^n C_i - \sum_{j=0}^{\mu'_n-1} D_j - K_n \cdot D_{\mu'_n} = MIN \quad 0 \leq K_n \leq 1 : 0 \leq \mu'_n \leq M \quad (4.24)$$

For each discharge, m , complementary constraints are applied. In that case each discharge m cannot end before and amount of charges has been made:

$$I + \sum_{i=0}^{v_m-1} C_i + \kappa_m \cdot C_{v_m} - \sum_{j=1}^m D_j = MIN \quad 0 \leq \kappa_m \leq 1 : 0 \leq v_m \leq M \quad (4.25)$$

In the same way, each discharge m cannot end after the contribution of charges given with the following equation has been done:

$$I + \sum_{i=0}^{v'_m-1} C_i + \kappa'_m \cdot C_{v'_m} - \sum_{j=1}^m D_j = MAX \quad 0 \leq \kappa'_m \leq 1 : 0 \leq v'_m \leq M \quad (4.26)$$

The identification of all the partial sequences leads to the addition of new links to the EON. Each final variation node is constrained at least between other two variation nodes. There are also links between charges and discharges to represent both sequences in the same way as the SIO case. The aspect of the EON focused on a charge is represented figure 4.26.

In the same way for each discharge there is a set of associated charges and constraints that can be translated in the EON representation as shown in figure 4.27.

4.5. Storage management

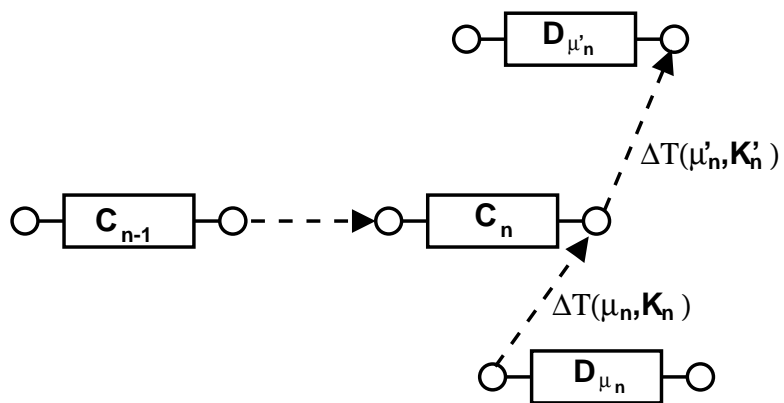


Figure 4.26: SISO constraints associated to a charge

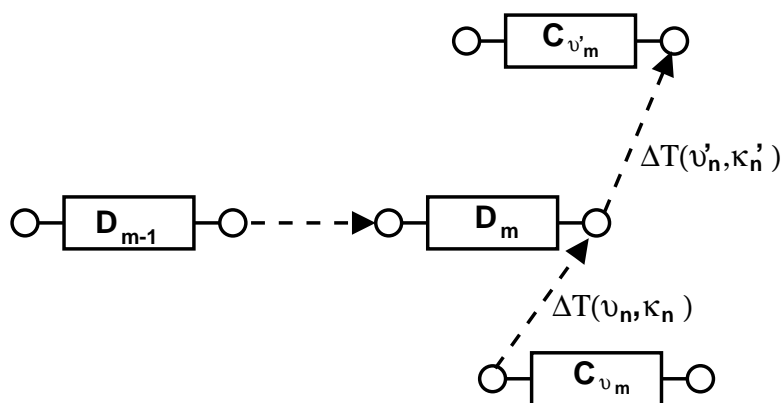


Figure 4.27: SISO constraints associated to a discharge

Chapter 4. The EON model

4.5.3 MIMO: Multiple Input Multiple Output

The methodology of the previous case can be generalized to the most complex situation. As it have been seen, in the SISO case there is no need to define links at the beginning of each operation because of the hard constraints that exist between the charges and the discharges. Nevertheless, in the general case as the constraints between charges and discharges does not exist, additional links are needed. That is, for each ending node of a variation, additional links are created to insure that at the end of the variation all the associated variations have also been done as is shown in figures 4.28 and 4.29.

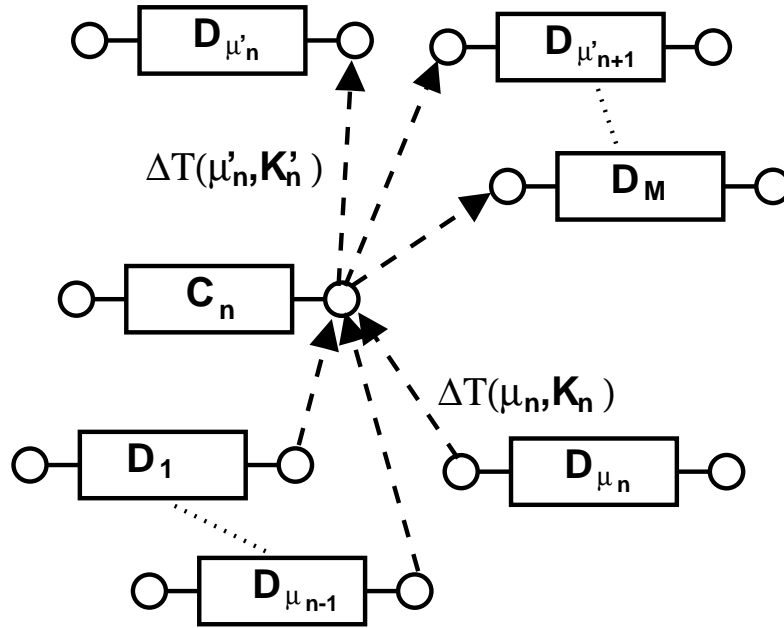


Figure 4.28: MIMO constraints associated to an end of a charge

Additionally, new constraints have to be added to the beginning event of each variation. To insure that the levels of the storage will be in the limits established, the links to apply at the beginning node of each charge will be the same as the links applied at the ending node of the previous charge or discharge. So in order to keep the profile level between the maximum and the minimum level links must be established between the charge and the discharge sequences so that each charge n cannot begin before a contribution of discharges has been made according to the equation:

$$I + \sum_{i=1}^{n-1} C_i - \sum_{j=0}^{\mu_{n-1}-1} D_j - K_{n-1} \cdot D_{\mu_{n-1}} = MAX \quad 0 \leq K_{n-1} \leq 1 : 0 \leq \mu_{n-1} \leq M \quad (4.27)$$

4.5. Storage management

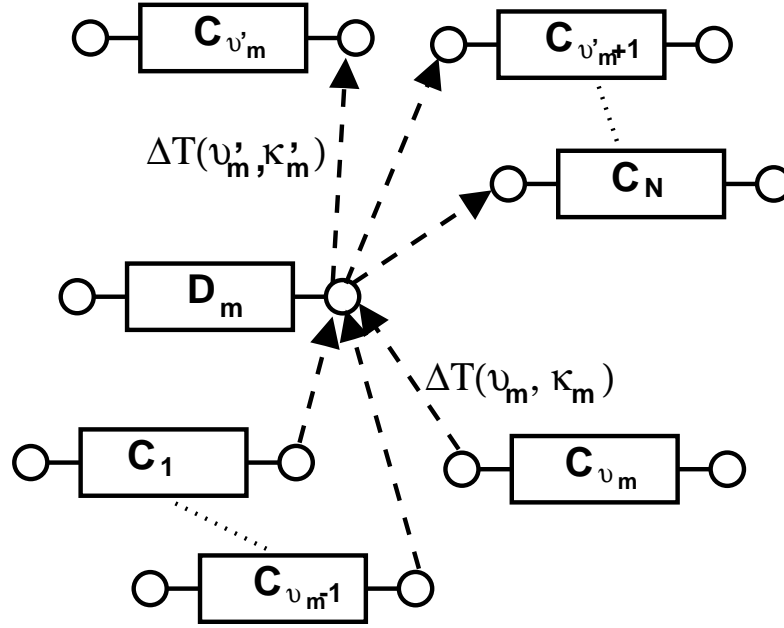


Figure 4.29: MIMO constraints associated to an end of a discharge

and also cannot begin after the contribution of discharges given with the following equation has been made:

$$I + \sum_{i=1}^{n-1} C_i - \sum_{j=0}^{\mu'_{n-1}-1} D_j - K_{n-1} \cdot D_{\mu'_{n-1}} = MIN \quad 0 \leq K_{n-1} \leq 1 : 0 \leq \mu'_{n-1} \leq M \quad (4.28)$$

For each discharge, m , complementary constraints are applied. In that case each discharge m cannot start before an amount of charges has been made according with the equation:

$$I + \sum_{i=0}^{v_{m-1}-1} C_i + \kappa_{m-1} \cdot C_{v_{m-1}} - \sum_{j=1}^{m-1} D_j = MIN \quad 0 \leq \kappa_{m-1} \leq 1 : 0 \leq v_{m-1} \leq M \quad (4.29)$$

and also, each discharge, m , cannot start after the contribution of charges given with the following equation has been done:

$$I + \sum_{i=0}^{v'_{m-1}-1} C_i + \kappa'_{m-1} \cdot C_{v'_{m-1}} - \sum_{j=1}^{m-1} D_j = MAX \quad 0 \leq \kappa'_{m-1} \leq 1 : 0 \leq v'_{m-1} \leq M \quad (4.30)$$

Chapter 4. The EON model

As in the previous case the identification of the sequences for every charge and every discharge leads to the addition of links to the EON. The aspect of these links related to the beginning time of a variation is shown in figures 4.30 and 4.31.

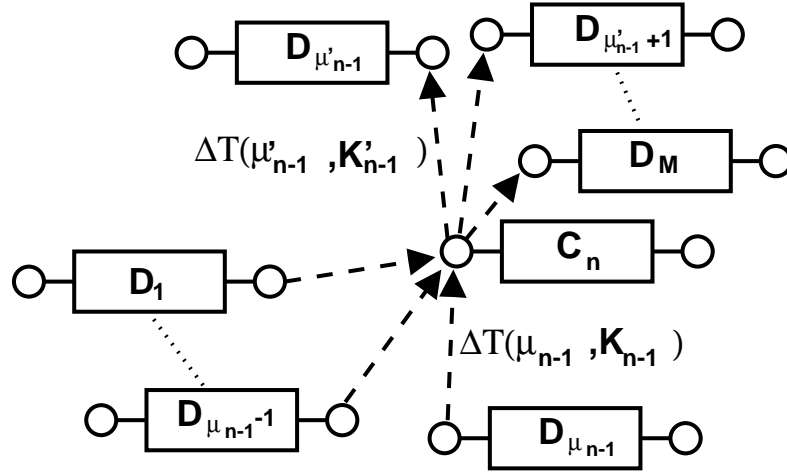


Figure 4.30: MIMO constraints associated to the start event of a charge

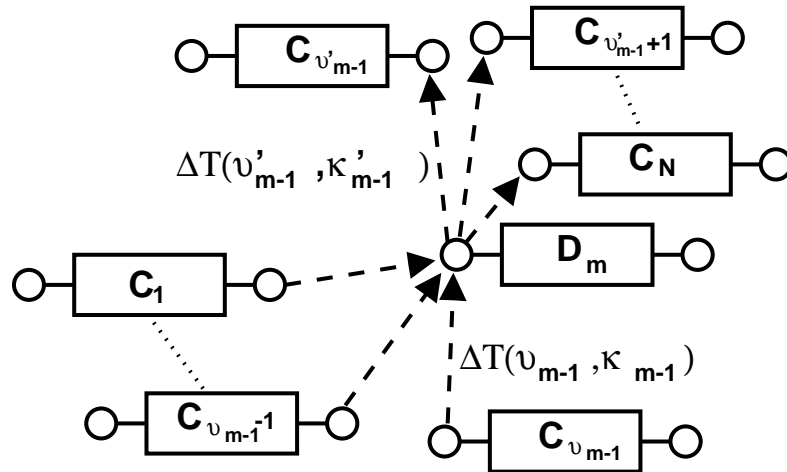


Figure 4.31: MIMO constraints associated to the start event of a discharge

4.6. Solving the EON model

4.5.4 SIMO:Single Inlet and Multiple Outlets

This case is a combination of the two previous cases SISO and MIMO. The charges will have the links of the SISO case and the discharges will have the links of the MIMO case. Other way to solve this case is to use all the MIMO links plus additional links from the end event of a charge to the start event of the following one in the sequence.

4.5.5 MISO Multiple Inlet and Single Outlet

This case is also a combination of the two previous cases SISO and MIMO. The charges will have the links of the MIMO case and the discharges will have the links of the SISO case. Other way to solve this case is to use all the MIMO links plus additional links from the end event of a discharge to the start event of the following one in the sequence.

4.5.6 Test case

To illustrate the storage model a simple test case has been designed. The storage constraint has been included into a general scheduling tool based in the EON (Appendix A). Two simple recipes have been described in the hierarchical form and introduced as an input data. Each recipe has one task contemplating two operations; one discontinuous and the other semi-continuous. The semi-continuous operation generates or consumes the up-stream product material as one recipe generates this material and the other recipe uses it as a raw material. There are four processing units available.

The first recipe can be assigned to the two first process units which have a capacity of 25000 units of production for each batch. The second recipe can be processed in Process unit three and four which have a capacity of 10000 production units for each batch. The capacity of the intermediate storage has been designed for 40000 units and there is an initial level of 10000 units. Without applying the storage constraints the aspect of the gantt chart and the storage profile is shown in figure 4.32.

This situation shows clearly the kind of problems that need to be solved. The first model to be tried is the SIO model, once applied the additional constraints are generated into the EON. The solution of the modified EON can be appreciated in figure 4.33.

The application of the SISO model decreases drastically the makespan due to the possibility of charge and discharge at the storage at the same time. As is shown in figure 4.34.

The application of the MIMO model instead of the SISO model does not affect the makespan but leads to an important change in the evolution of the level in the storage due to the different constraints as is shown in figure 4.35.

The application of the SIMO model leads up to the same solution as the SISO model because of the recipe constraints related to the second recipe leads up to the constraint of no overlapping of discharges. Otherwise, the application of the MISO model generates for this case the same solution that has been generated with the MIMO for the same reason as before.

4.6 Solving the EON model

The EON models have been solved by two different methods:

Chapter 4. The EON model

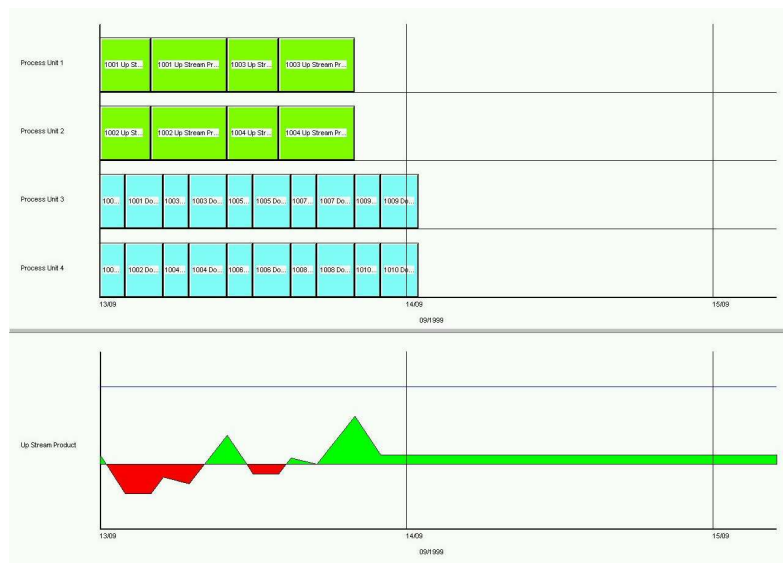


Figure 4.32: Solution of the EON without storage constraints

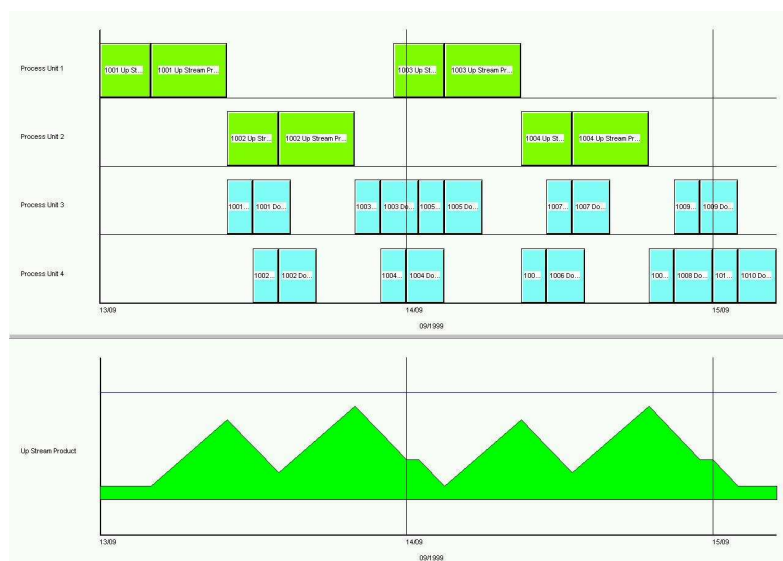


Figure 4.33: Solution of the EON with SIO constraints

4.6. Solving the EON model

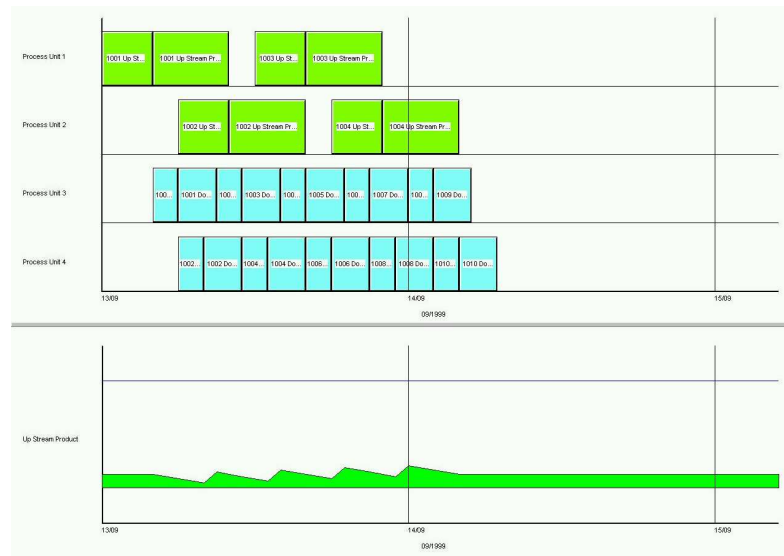


Figure 4.34: Solution of the EON with SISO and SIMO constraints

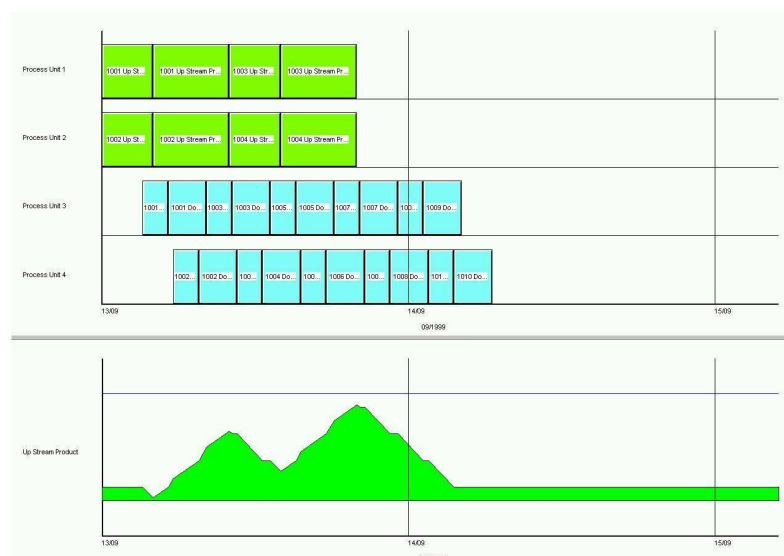


Figure 4.35: Solution of the EON with MIMO and MISO constraints

Chapter 4. The EON model

- An standard method using directly the LP formulation of the EON model
- A graph based method using the properties of the Event operation Networks

4.6.1 Solving EON using LP formulation

This was the first method used for solving the EON graphs generated as it is derived directly from the formulation.

As it was shown on page 41 the final LP formulation for the EON problem is the following one:

minimize:

$$Z = C^1 \cdot MS + \sum_{m=1}^M C_m^2 \cdot TW_m + \sum_{n=1}^N C_n^3 \cdot T_n \quad (4.31)$$

subject to:

$$0 \leq MS \geq T_n \quad \forall n; n = 1 \dots N \quad (4.32)$$

$$0 \leq T_n \geq T_n^{\min} \quad \forall n; n = 1 \dots N \quad (4.33)$$

$$T_{FN_m} - T_{IN_m} - TOP_m = TW_m \quad \forall m; m = 1 \dots M \quad (4.34)$$

$$0 \leq TW_m \leq TW_m^{\max} \quad \forall m; m = 1 \dots M \quad (4.35)$$

$$T_{DN_k} \geq T_{ON_k} + \Delta T_k \quad \forall k; k = 1 \dots K \quad (4.36)$$

Once a EON description of a problem is given, the construction of the LP associated problem using equations from 4.31 to 4.36 is obvious and can be fully automated. Once the problem is built, it can be easily solved using any of the LP-solver available packages.

The objective function (4.31) requires a set of different weights C^1, C_m^2 , and C_n^3 where C^1 is the most important weight as it implies the makespan (MS). The other sets of weights are included to avoid possible problems of degeneracy in the solution, specially, when the different TW_m can accept values different from zero. Normally the weights values of C_m^2 , and C_n^3 will be much lower than the main weight (C^1)

4.6.2 Solving EON using FCEON algorithm

Solving an LP problem is not the only way to obtain the solution of a given EON structure. It is possible to solve a EON graph using an alternative way that requires much less resources and memory than the required by standard methods of solving LP problems. This method has two parts: a main algorithm that explores and adjusts the times of the nodes going forward along the net and an auxiliary recursive algorithm that adjusts the times of the nodes backwards to insure that all the constraints are accomplished. The main algorithm can be summarized as follows:

4.6. Solving the EON model

```

Calculate EON
do
    for each node in the net

        if all the precedent nodes have been calculated
            Calculate a new  $T_n$ 
            for each precedent node
                if a constraint is violated
                    calculate backwards
                endif
            endfor
        endif

    while there is a node to calculate and new nodes have
    been calculated in the last iteration
        if all the nodes has been calculated
            return solution
        else
            return problem infeasible
        endif
    endfor

```

The “calculate backwards” function is a recursive function that has the mission to adjust the different T_n to insure that all constraints are respected. This algorithm is described below:

```

Calculate Backwards
if a  $TW$  is possible
    calculate the necessary  $TW$ 
    if  $TW_{calculated} \geq TW^{max}$ 
        Calculate a new  $T_n$ 
    endif
else
    Calculate a new  $T_n$ 
endif
if  $T_n$  has been changed
    for each precedent node
        if constraint is violated
            calculate backwards
        endif
    endfor
endif

```

Chapter 4. The EON model

The calculation of T_n is easily performed. Each node n is bounded by a T_n^{min} , and additionally may have precedent nodes linked by equality constraints because of the operations like:

$$T_n = T_{n'} + TOP_m + TW_m \quad (4.37)$$

Which is really an inequality constraint, since TW_m is bounded by 0 and TW_m^{max} , so it is transformed to

$$T_n \leq T_{n'} + TOP_m + TW_m^{max} \quad (4.38)$$

$$T_n \geq T_{n'} + TOP_m \quad (4.39)$$

Besides, the precedent nodes linked between events generate inequality constraints like

$$T_n \geq T_{n'} + \Delta T_k \quad (4.40)$$

In order to calculate a T_n this algorithm transforms the equations 4.39 and 4.40 into the following equality constraints:

$$T_n = T_{n'} + TOP_m \quad (4.41)$$

$$T_n = T_{n'} + \Delta T_k \quad (4.42)$$

With all the equations of the kind 4.41 and 4.42 derived from the corresponding precedent nodes and the T_n^{min} bound of the node a set of possibles T_n is obtained. From that set the highest value is chosen and fixed as a new value for T_n . Of course, there is the possibility of violation of the constraints 4.38 and 4.39. This possibility is checked and in the case of a violation a backward adjustment of the precedent nodes is performed.

4.6.2.1 Example

It follows a motivating example that shows how this algorithm works. A detailed step by step solution of the EON structure represented in figure 4.36 will be described.

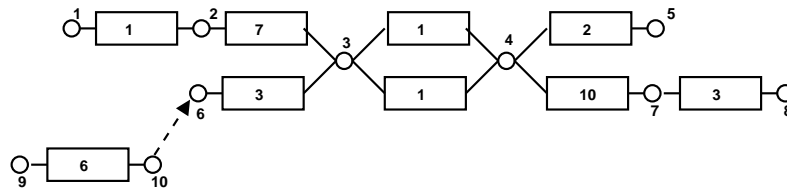


Figure 4.36: EON graph to be solved

This EON network has 10 events which are numbered in the figure. The objective is to find the associated T_n for each of these events. For sake of simplicity, all the operations have a TW_m^{max} equal to 0. TOP_m is shown in the figure as the value inside each operation. The T_n^{min} for each node is equal to 0. The summary of the evolution of the T_n when the algorithm is performed is summarized in the table 4.1 on page 70.

4.6. Solving the EON model

- First iteration
 - Node 1 has no predecessors, the value of T_n^{min} of this node is 0 so the value of 0 is assigned to this node.
 - Node 2 has node 1 as predecessor which has already a value assigned. The associated constraint is $T_2 = T_1 + 1$. The value of $T_1 = 0$. Therefore the value of T_2 is 1 so the value of 1 is assigned to node 2.
 - Nodes 3 to 8 cannot be calculated because they have a predecessor that have not been calculated. Node 9 does not have any predecessor, since the value of T_n^{min} of this node is 0, the value of 0 is assigned to this node 9.
 - Node 10 can now be calculated because node 9 has a value. The associated constraint is $T_{10} = T_9 + 6$ then the value of T_{10} should be 6 so the value of 6 is assigned to node 10.
 - There are no more nodes, to explore as there are nodes which do not have any value then a second iteration is performed.
- Second iteration
 - Nodes 1 and 2 are calculated and 3,4, and 5 cannot be calculated because they have some of the predecessors with no value assigned. Node 6 can be calculated because node 10 has a value assigned. The associated constraint to this node is $T_6 \geq T_{10}$ since the value of T_{10} is 6 and the criteria established is to start all the operations as soon as possible, the value of 6 is assigned to node 6.
 - Nodes 7 and 8 cannot be calculated due to the lack of values for node 4, so as there are nodes which do not have any value and in this iteration a new node has a value assigned, a third iteration is performed.
- Third iteration
 - Nodes 1 and 2 have already a value, Node 3 can be calculated since there is a value for node 2 and 6, the associated constraints are $T_3 = T_2 + 7$ and $T_3 = T_6 + 3$, as $T_2 = 1$ and $T_6 = 6$ these constraints end up with a value of 8, the first one and 9 the second one. The highest value (9) is chosen and set in T_3 . The backward checking shows that some constraints of precedent nodes are violated, so backward propagation is needed for node 2.
 - Backward calculation of node 2 is performed. The constraint to be satisfied is $T_2 = T_3 - 7$ as the value of T_3 is 9 then, the new value of T_2 is 2. The backward checking shows that some constraints of precedent nodes are violated, so backward propagation is needed for node 1.
 - Backward calculation of node 1 is performed. The constraint to be satisfied is $T_1 = T_2 - 1$ as the value of T_2 is 2 then, the new value of T_1 is 1. The backward checking shows that there are no more constraints to be satisfied, so calculation continues.

Chapter 4. The EON model

- Node 4 can be calculated, the constraint associated is $T_4 = T_3 + 1$ (for both operations) as the value of $T_3 = 9$ then the value of 10 is assigned to T_4 .
- Node 5 can be calculated, the constraint associated is $T_5 = T_4 + 2$. The value of T_4 is 10, so the value of 12 is assigned to T_5 .
- Node 6 is already calculated and node 7 can now be calculated. The constraint associated to node 7 is $T_7 = T_4 + 10$ as $T_4 = 10$, the value of T_7 will be 20.
- Node 8 can be calculated, the constraint associated is $T_8 = T_7 + 3$, $T_7 = 20$ so the value of 23 is assigned to T_8 .
- Nodes 9 and 10 are already calculated, as all the nodes has been calculated the algorithm ends without error.

Table 4.1: Evolution of the values of T_n along the different iterations of the algorithm

	first iteration					second iteration	third iteration						
node/step	0	1	2	3	4	1	1	2	3	4	5	6	7
1	-	0	0	0	0	0	0	0	1	1	1	1	1
2	-	-	1	1	1	1	1	2	2	2	2	2	2
3	-	-	-	-	-	-	9	9	9	9	9	9	9
4	-	-	-	-	-	-	-	-	-	10	10	10	10
5	-	-	-	-	-	-	-	-	-	-	12	12	12
6	-	-	-	-	-	6	6	6	6	6	6	6	6
7	-	-	-	-	-	-	-	-	-	-	-	20	20
8	-	-	-	-	-	-	-	-	-	-	-	-	23
9	-	-	-	0	0	0	0	0	0	0	0	0	0
10	-	-	-	-	6	6	6	6	6	6	6	6	6

4.6.3 Comparison between the two methods

A comparison between both methods of solving the EON graphs has been performed. Intuitively, the second method should be faster than the classical methods for solving LP problems since it is based directly in the graph representation. Otherwise, the LP formulations allows a better control upon the objective function than the algorithm proposed that has the objective to set all the T_n as soon as possible.

In order to check the performance of both methods, several experiments has been carried out. As an illustrative example of the results the consumer products test case has been chosen (the description of the test case can be found in section 7.3). Several schedules between 50 and 1000 batches has been generated and the timing has been calculated using both methods.

Figure 4.37 shows the CPU time required for both methods. The hardware used was a 2 GHz Athlon AMD processor running under Windows 2000. LP models were solved using GAMS/CPLEX whilst FCEON was coded in C++. As mentioned earlier, the CPU time

4.7. Conclusions

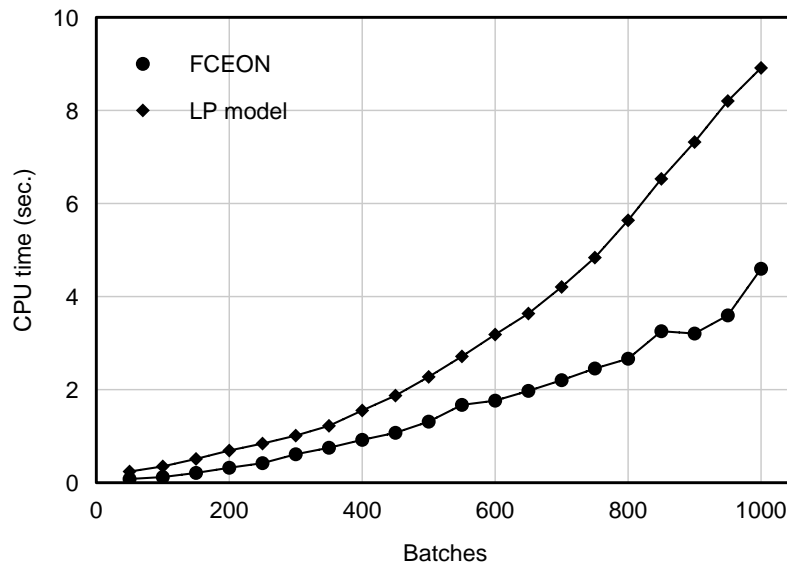


Figure 4.37: CPU time used in the two EON solving methods presented

required for solving the LP formulation of the EON timing model is twice the CPU time required by solving the same model using the FCEON algorithm. In the other hand, both methods reach the same solution and require small CPU times with big problems.

4.7 Conclusions

In this chapter the EON model for operations timing has been presented. The main strength of EON model is that it allows the representation of complex time constraints between operations in the same schedule using simple components as the events, operations and links.

The construction of the EON for different kinds of links has been discussed as well as a methodology for the representation of storage constraints as time constraints.

An iterative procedure that allows to take into account limited resources has also been described.

Finally, two different methods for calculating the EON method has been shown and compared.

Chapter 4. The EON model

Nomenclature

C_i : Charge (level increase) of an storage

D_i : Discharge (level decrease) of an storage

DN_k : Destination event of link k

ET' : Finishing time of operation'

ET_n : Finishing time of operation n

FN_m : Final event of operation m

IN_m : Initial event of operation m

IT' : Starting time of operation'

IT_n : Starting time of operation n

K : Delta time

K_n : Fraction of discharge μ_n

MAX : Maximum level associated to an storage

MIN : Minimum level associated to an storage

MS : Makespan value

NI_{V_n} : Initial event of variation V_n

NF_{V_n} : End event of variation V_n

ON_k : Origin event of link k

TOP_n : Operation time associated to the operation n

TW_m : waiting time of operation m

TW_m^{max} : Maximum waiting time for operation m

$TWmax_n$: Maximum waiting time for the operation n

T_n : Time value associated to event n

T_n^{min} : Minimum time associated to event n

V_i : Level variation of an storage

Z : Objective function value

ΔT_k : Delta time associated to link k

4.7. Conclusions

κ_m : Fraction of charge v_m

μ_n : Discharge index associated to charge n

v_m : Charge index associated to discharge m

Chapter 4. The EON model
